

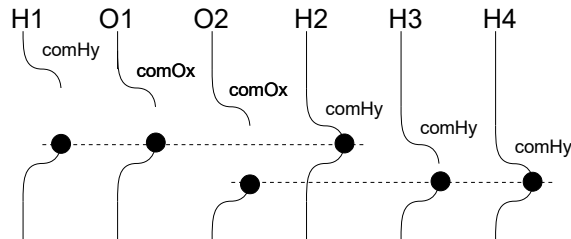
CC4302 Sistemas Operativos – Tareas 4 y 5 – Semestre Primavera 2023 - Prof.: Luis Mateu

En estas 2 tareas Ud. deberá implementar una nueva herramienta de sincronización para nThreads basada en la formación de moléculas de agua. Los encabezados de la funciones pedidas son:

```
typedef struct h2o *nH2o;  
nH2o nCombineOxy(void *oxy, int timeout);  
nH2o nCombineHydro(void *hydro);  
void initH2O(void);  
void endH2O(void);
```

La función *nCombineOxy* suministra un átomo de oxígeno (*oxy*) y *nCombineHydro* suministra un átomo de hidrógeno (*hydro*). En cuanto haya una llamada pendiente de *nCombineOxy(o)* y 2 llamadas pendientes de *nCombineHydro* con átomos *h1* y *h2*, Ud. debe crear un objeto de tipo *nH2o* invocando la función *makeH2o(h1, h2, o)*. La función *nCombineOxy* espera por a lo más *timeout* milisegundos retornando NULL si expiró. Si *timeout* es -1 se espera indefinidamente. Por simplicidad, *nCombineHydro* no tiene *timeout*.

El siguiente diagrama de threads muestra cómo se deben asociar las llamadas de *nCombineOxy* (abreviado *comOx*) con las llamadas de *nCombineHydro* (*comHy*).



Tarea 4: Implemente las funciones solicitadas considerando que el parámetro *timeout* es siempre -1 (es decir sin *timeout*).

Tarea 5: Implemente las funciones solicitadas sin restricción alguna sobre el parámetro *timeout* (es decir con *timeout*).

Restricciones

Ud. debe programar las funciones solicitadas como herramientas de sincronización nativas de nThreads, es decir usando operaciones como *START_CRITICAL*, *setReady*, *suspend*, *schedule*, etc. Ud. no puede implementar la API solicitada en términos de otras herramientas de sincronización pre-existentes en nThreads (como semáforos, mutex, condiciones o mensajes). Ejemplos de la solución que se espera de Ud. son la implementación de los semáforos, mutex, condiciones y mensajes

de nThreads (en los archivos *nKernel/sem.c*, *nKernel/mutex-cond.c* y *nKernel/nmsgs.c*).

Instrucciones

Descargue *t4.zip* de U-cursos y descomprímalo. Ejecute el comando *make* sin parámetros en el directorio *T4* para recibir instrucciones acerca del archivo en donde debe programar su solución (*T4/h2o.c*), cómo compilar y probar su solución, los requisitos que debe cumplir para aprobar la tarea y cómo entregar su tarea por U-cursos. Además se explica cómo puede probar sus tareas 1, 2 y 3 usando nThreads como implementación de pthreads.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *h2o.zip* generado por *make zip*. Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos. Solo así estará seguro de no haber entregado archivos incorrectos. Se descuenta medio punto por día de atraso. No se consideran los días de receso, sábado, domingo o festivos.